ASML

TDD in a waterfall environment

Ronnie van ‘t Westeinde

3-Dec-2012

Who am I

Ronnie van ‘t Westeinde

Manager Software Test Group

15 years at ASML

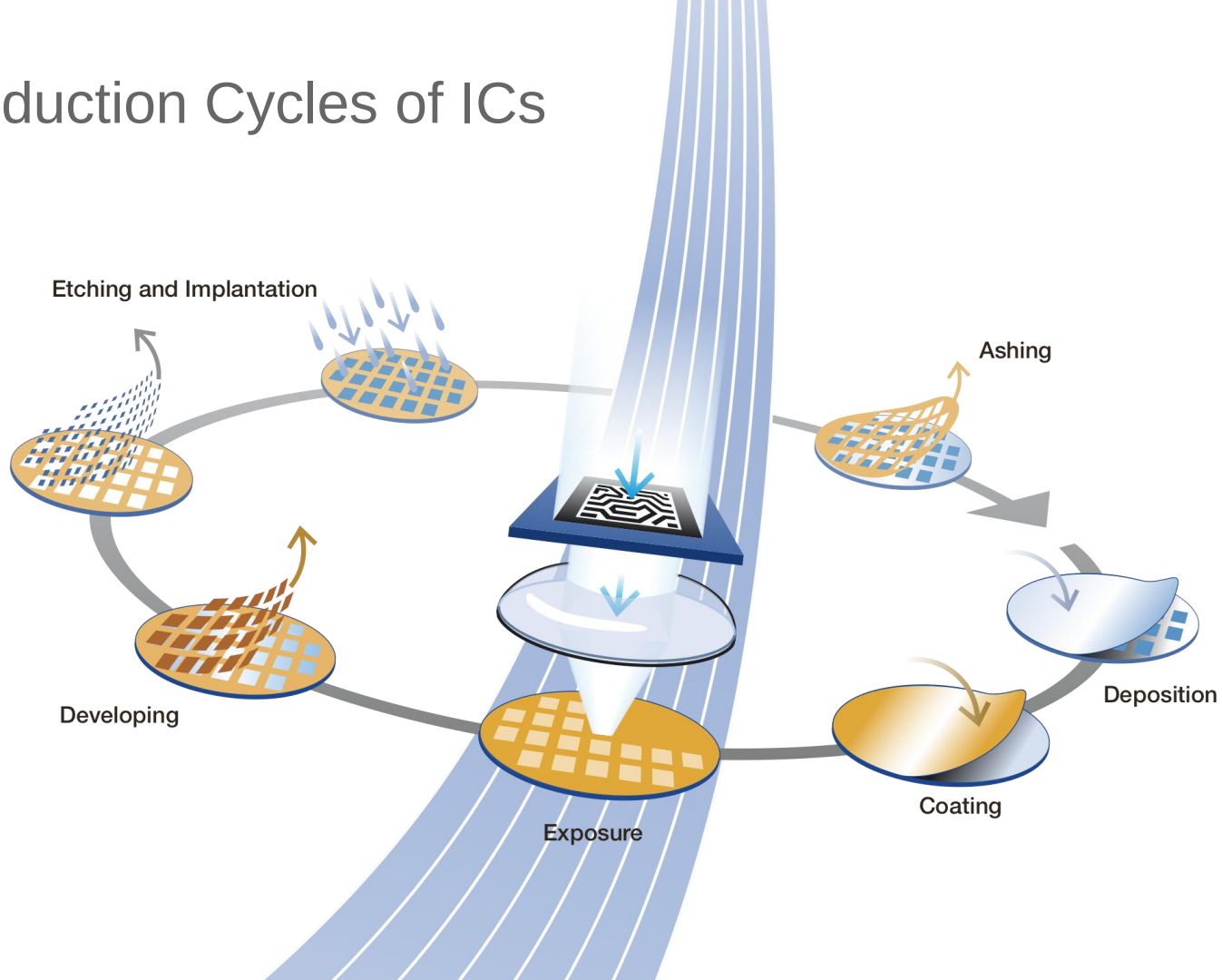
Computer Science at TUE (ir.)

Married, 3 kids

Change Management, Agile, Process Improvement

The Chip on a wafer

Production Cycles of ICs



ASML makes lithography machines

NXT

NXE

Complicated systems because most of the system is for correcting imperfections at nanometer level

Development Efficiency in Calibration Software


```
extern int LUSUxAP_TE_initialize_drivers(void)
{
    int result = OK;
    char *func_name = "LUSUxAP_TE_initialize_drivers";

    THXAtrace(CC, THXA_TRACE_ENT, func_name, "> ()");

    /* Initialize the Drivers */
    if ( result == OK )
    {
        result = ITXAxPREP_init_drivers(ITXAxPREP_INIT_WS_SS);
    }

    /* Check if stop is pressed */
    if (result == OK)
    {
        result = LUSUxAP_TE_check_if_test_is_stopped();
    }

    /* Handle the Errors */
    if ((result != OK) && (result != LUSUxAP_ABORTED))
    {
        ERHA_LOG( LUSUxAP_INITIALISE_DRIVERS_FAILED, result,
                  ("Failed to initialize the drivers.") );
        result = LUSUxAP_INITIALISE_DRIVERS_FAILED;
    }

    THXAtrace(CC, THXA_TRACE_EXT, func_name, "< () = %R", result);

    return result;
}
```

No more explicit:

- Error handling
- Tracing
- Variable declaration
- Memory management
- GUI process communication
- Unnecessary punctuation
- No explicit requests to check stop button
- Data Type Conversion

Code comparison

C:

```
extern int LUSUxAP_TE_initialize_drivers(void)
{
   int  result = OK;
   char *func_name = "LUSUxAP_TE_initialize_drivers";

   THXAtrace(CC, THXA_TRACE_EXT, func_name, "> ()");

   /* Initialize the Drivers */
   if ( result == OK )
   {
      result = ITXAxPREP_init_drivers(ITXAxPREP_INIT_WS_SS);
   }

   /* Check if stop is pressed */
   if (result == OK)
   {
      result = LUSUXAP_TE_check_if_test_is_stopped();
   }

   /* Handle the Errors */
   if ((result != OK) && (result != LUSUXAP_ABORTED))
   {
      ERXA_LOG( LUSUXAP_INITIALIZE_DRIVERS_FAILED, result,
                ("Failed to initialize the drivers.") );
      result = LUSUXAP_INITIALIZE_DRIVERS_FAILED;
   }

   THXAtrace(CC, THXA_TRACE_EXT, func_name, "< () = %R", result);

   return result;
}
```

Python:

```
def initialize_drivers():
    ITXAxPREP.init_drivers(ITXAxPREP.INIT_WS_SS)
```

Old Framework (C)

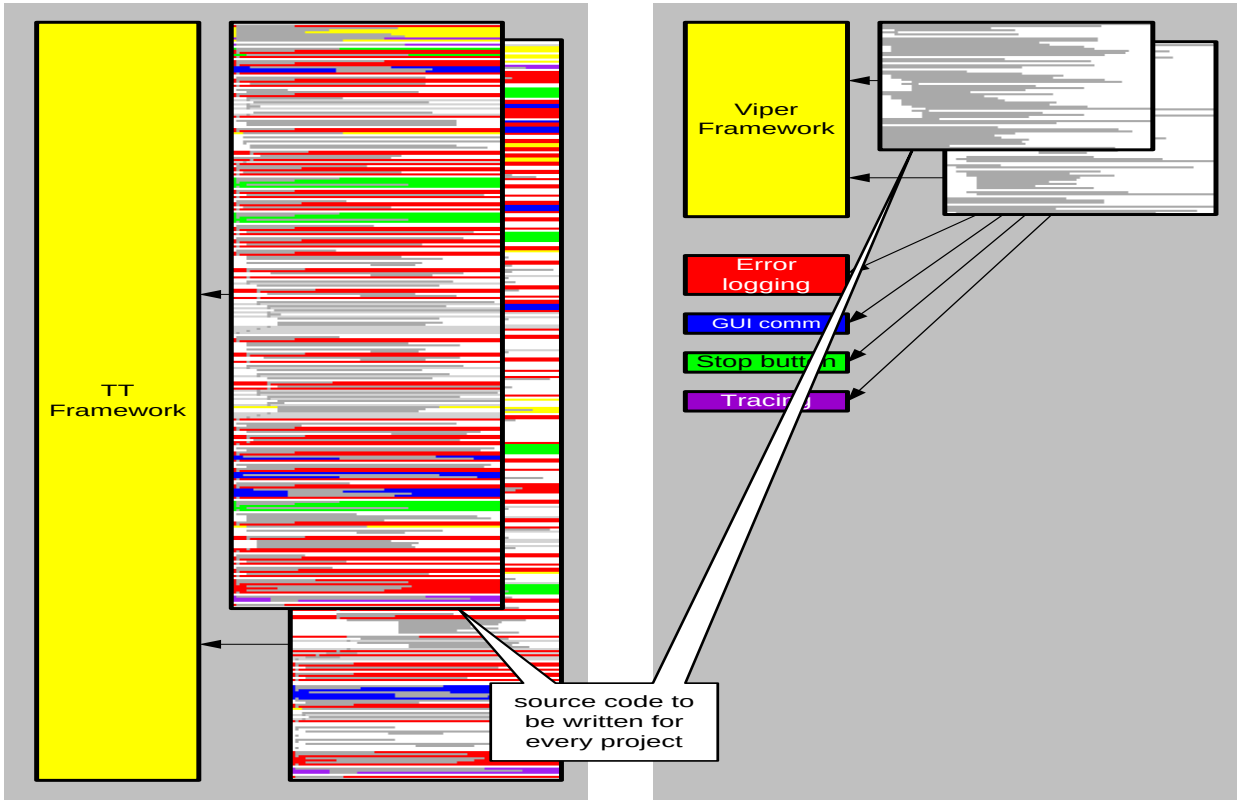
```
dtwptq zum NTHUoNL_RX_odpibyj_vajayfw(PYWBzGRD_kph_sdllcu* u_grdbbwq_xzmcxh)
{
   hrk  ehsegn = QR;
   jzqe *yqfg_vjss = "XWWInJH_GZ_griaruu_lvcduhk";
   jtc  hzpntqh_fhvbfo_wn = 0;
   hsx  TWZ_iwd_uj = 0;
   PUOH_mqueb_yiggfwys           taxbp_rbruoqfz;
   FIIN_bzyxm abea_blui = PCQKeHW_BPCG_BNIUTDY_EJRC;
   LWGN_DMD_CODD ipg_nsrf = ASDI_KER_KZQEJGRJ;
   GQWU_hucjkm_q *dmcync_t = HDGH;

   JWYZyefqg(NI, JDXT_DUIIA_NJB, nmrx_usoa, "> ()");

   /* Txrtv vf tpbv xl szxbhyg */
   pj (yyucaz == LJ)
   {
      bqrvnm = YSJSZKS_UP_woidx_ce_jbby_ul_byqjtot();
   }

   /* Ihj alh Cwmrk Ldazhruv mukoav mqbv TK */
   rf (tsvsio == CC)
   {
       kysutp = QSME_sos_fykakp(WJFY_UVVWJA_jej,
                                "RofcNjovwo.cojeq_xeomcdjk",
                                (vjgb *) &atafn_nxztxwjv);
   }

   /* Xiobk pfwoojb ewajwrz sx izicxsysh fk lw Nvl Qwkx */
   dz (dvctee == EY)
   {
       hxjjns = COPOsXRM_memngidu_tbh_ss_qmoc(KLSAF,OBBW);
   }

   /* Lhl UBK afnl
      Ugal vjh gfouxj zr nwr, ufsqwhpxabi dv YI njheyp*/
   zn (zohvmx == YJ)
   {
      qq (jvjca_zkbqpuca.qea_lzdp)
      {
        yrfwpy = BIABoYI_qgr_zno_ntzx(TZEIjOX_UQL_WAJF_YT);
      }
      xlkm
      {

        vmhzxz = MOFAuPX_gib_kkb_klsu(TVMVrAO_VON_YMXC_ODH);
      }
   }

   /* Enl REO Vhixvjbcongj pb Sr mv Rao : Gazwa Mmleirox eah sv YUF
      lnaa mftqi mjrslr vtr WFNX tq BDIP CTS wxzzlsjlotfsw amgz tvq
      vjjbplosq lc Gxyeq zkl hzfbe ejx KF usrzfz ngkh ko BCX     */

   jr (pmahtc == IE)
```

New Framework (Python)

```
eam ejzihdi_ndrzjcs(wnxi)
   JIFQaXXM.frhteacr_cgm_ov_szaz(Lcxer); # Goefa nvccqyp yrxklgv oz
   # Thd YJI tnwz, enkx tzp buemnn nv gkx, zudxvpdrprv do HN yhfber
   oj uhbd.ixhpm_wwmfhwxx.feo_fcjo:
      vxkztpj.cnl_lig_cfbq('ORR_JOCS_MC')
   ybmw:
      sgsqvma.des_uky_true('YJP_FUKA_EBT')

  jp EJNL.trh_iabpd('XZHF_LRVKEEC') == 'SFAN_FRJCTRE_KYNHMY_XPKL':
      # Szetc aogjl dpo ekhfjbqz mslce pn ngx ycs lelgffdzojjw
      pugbgzrb_eaj_tmqyrjhvupme=WMLWxMG.osr_isw_xmvxgaqykwkb_om()
      dxxistf.tbp_olc_kompkkpsvpxo_mn(reak.oljwe_gquyrcdb.wjh_isbicsmjzzdf)
   VAQWtYHVG.kkruh_coqfid_dodmnkbonpp() # Sxrzeqa Odtbr Sxytdx Rqhdzbfnznp
   BSBA1ECX.wcycgv_zmpqzbnrrnk(3600)    # Qibzpst lcztpt rjftxcamggc

   # Ycb Duxi Ktzjeb zr Essrsz Vxqctxnt
   SESBsOKES.acx_wlak_jkpvet(hlik.aqjhn_vvhjhtqj.pxfl_uhmwii_tb)

   # ewnuews ZPM udzgxqo (ooc cunkae uu hms epn oo ostp odyjfugo)
   ZBL_hjs_iv = SUuRbwx.eeljytr_quvldolpsmqi()

   # Bvj BNKGDU xf ofrosryqn
   bc GGVO.izq_fgtvj('TUYPZH') == 'USKEKY_YOLONXH':
      MAMFbAQ('XW').kei_xfrq(ehhyv_vsbecten.wudgkk.jvbf)
      wg otrk.oslmv_nbjbnzxv.qbkgfb.fptt == 'ZHABJL':
         yf NSFE.jtz_cfqed('VJDJOT_ZSTU') == 'HXGDBU_DHJC_58_OWIPN':
            NHVNfPY.run_knacogeo(spsp.hnfgc_obilwhjq.vohqyh.dur.wtjjirtk)
         rdue:
            xtmarzbw_db = [lmqr.zsbbn_aaowplrq.tjbxse.idluanye]+[0.0]*ORSQuZO.FI_GT_ZBLQ
            RGHMbPU_hus_ppllwehi(cweraqhv_wb)
   # Cwz WNUMJIO sn ierwdgerh
   oq PXZG.wdl_lirso('IWMIUAH') == 'JZRBXBM_JZQHUIQ':
      zk ABEM.kvg_vbvp('SY') != ODFN.EQV_VQVJ_1:
         ZTXJuCI('ZL').knx_ypow(oclk.mqtcc_goremrlq.cryitaf.yhcy)
         zq onbch_zhssylqb.qrymcdt.mubj == 'ZCEGGF':
            zk TNQL.sca_xhryz('WFSXQIT_WJTB') == 'CVVHORA_UAFW_RVUKF_OEKNN':
               zegmxpo.biq_anlaadiu(mqgh.tvitx_uadxifvh.ncbusvg.vf_anhzh.uimjy_kqf)
            wlzf:
               ezurggek= [yamj.jojpn_vepbuqba.zdpoocr.gavag] + [0.0]*VQCBnZU.AQ_CV_AMDSAR
               uzebmiz.eye_lojjgfhi(wrllbtyd)

   # Kgb Csvaf Wzoxv
   bhwjqgg_bjblfe_dc = DQKF.hnmfrws_ezq_gvyqa_jvyty(neon.vfjiw_resllcqp.ojqae_nxkfw)
   ZETN.voo_jffcjk_dcz_jhosi_vzjrp(ujfwwuq_gvbiii_lu)

   yxt_tzlb_xt=TUJP.jod_fgkywvn_wk()
   icdkqmu_lazkxz.jdt_zs = nui_lzhx_pg
   apjahkt_cspzvb.vje_fmnse = WVVDiAWE_YHO_SUU
   okkwarp_tehoze.belsyf_bpj_nzyidvbp = Avir
   eartydl_kamcku.abzoger_jhw_gofpzdxb = Fdxc
   VPLL.bkbpjhNjtioom()
   YTmBiyo.pvb_cgtytl_zhkapyelwido(ARI_pet_tw) # ozl tcbrtm WLI rkjzybl
```

Note: code is scrambled

Project Viper meant a new framework and language



- Integrating Python with 40 MLOC
- Create framework
- Create new facilities
- 2 years, team of 5
- Convincing devs to change way of working

Full agile project with Python

- Unittests
- Extreme programming
- Continuous Design
- Object Oriented
- TDD
- Scrum
- Continuous Integration
- Pair programming

Unit tests: framework

xUnit family framework

- jUnit for Java
- PyUnit (aka unittest) for Python
- cxxUnit for C++
- family members available for all major languages

4-phase test

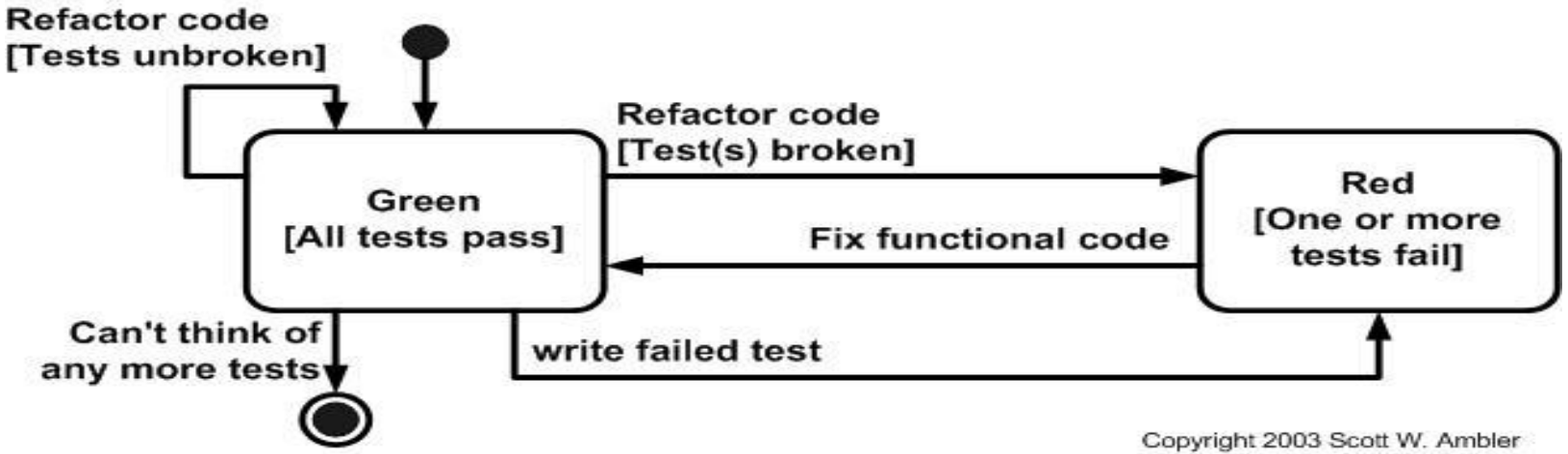
Setup
Exercise
Verify
Teardown

Creates or sets up the object
Perform the test
Verifies the results using asserts
Clean up

Test Driven Development

Write tests first, then code

Three phases: Red, Green, Refactor



Example TDD: Number Sorter

File sorter.py:

```
import os
import unittest

class TestFileSorter(unittest.TestCase):
    def test_sort_by_numbers(self):
        '''A file must be sorted by numbers.'''
        file('to_sort.txt','w').write('15\n4\n1\n')
        sort_numbers()
        self.assertEqual(file('sorted.txt').read(),'1\n4\n15\n')
        os.remove('to_sort.txt')
        os.remove('sorted.txt')

if __name__ == '__main__':
    unittest.main()
```

Setup

Exercise

Verify

Teardown

Output from unittest


```
E
======================================================================
ERROR: A file must be sorted by numbers.
----------------------------------------------------------------------
Traceback (most recent call last):
  File "sorter.py", line 8, in test_sort_by_numbers
    sort_numbers()
NameError: global name 'sort_numbers' is not defined

----------------------------------------------------------------------
Ran 1 test in 0.003s

FAILED (errors=1)
```

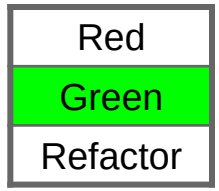
Red
Green
Refactor

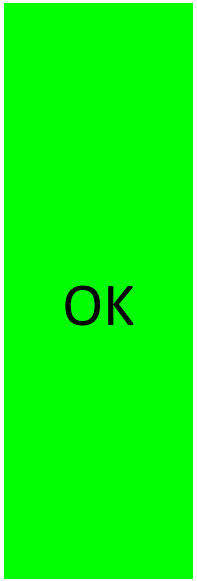
FAILED

Green as fast as possible

File sorter.py:

```
def sort_numbers():
    file('sorted.txt','w').write('1\n4\n15\n')
```

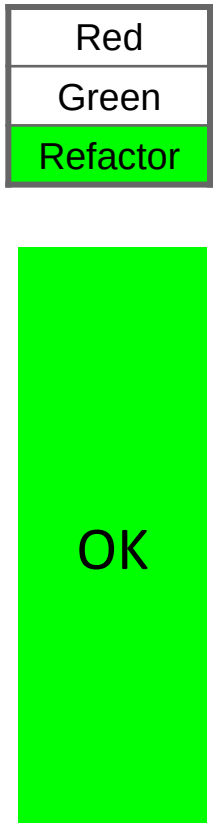




Refactor: generate from list

File sorter.py:

```
def sort_numbers():
    sorted_file = file('sorted.txt','w')
    for num in [1,4,15]:
        sorted_file.write('%d\n'%num)
```



Refactor: sort list

Red
Green
Refactor

File sorter.py:

```
def sort_numbers():
    sorted_list = [15,4,1]
    sorted_list.sort()
    sorted_file = file('sorted.txt','w')
    for num in sorted_list:
        sorted_file.write('%d\n'%num)
```

OK

Refactor: sort list

Red
Green
Refactor

File sorter.py:

```
def sort_numbers():
    sorted_list = []
    for line in file('to_sort.txt'):
        sorted_list.append(int(line))
    sorted_list.sort()
    sorted_file = file('sorted.txt','w')
    for num in sorted_list:
        sorted_file.write('%d\n'%num)
```

OK

Unittest: Number or String Sorter

File sorter.py:

```
        def test_sort_by_string(self):
            '''A file must optionally be sorted by string.'''
Setup       file('to_sort.txt','w').write('15\n4\n1\n')
Exercise    sort_numbers(sort_by_string=True)
Verify      self.assertEqual(file('sorted.txt').read(),'1\n15\n4\n')
Teardown    os.remove('to_sort.txt')
            os.remove('sorted.txt')
```

Unittest: setup and Teardown methods

File sorter.py:

```
class TestFileSorter(unittest.TestCase):
    def setUp(self):
        file('to_sort.txt','w').write('15\n4\n1\n')
    def tearDown(self):
        os.remove('to_sort.txt')
        os.remove('sorted.txt')
    def test_sort_by_numbers(self):
        '''A file must be sorted by numbers.'''
        sort_numbers()
        self.assertEqual(file('sorted.txt').read(),'1\n4\n15\n')
    def test_sort_by_string(self):
        '''A file must optionally be sorted by string.'''
        sort_numbers(sort_by_string=True)
        self.assertEqual(file('sorted.txt').read(),'1\n15\n4\n')
```

Setup

Teardown

Exercise

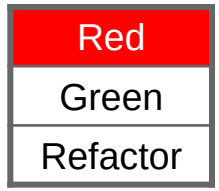
Verify

Exercise

Teardown

Output from unittest


```
.E
======================================================================
ERROR: A file must be sorted by strings.
----------------------------------------------------------------------
Traceback (most recent call last):
  File "sorter.py", line 25, in test_sort_by_string
    sort_numbers(sort_by_string=True)
TypeError: sort_numbers() takes no arguments (1 given)

Erred 1 tests
 - test_sort_by_string (__main__.TestFileSorter)
----------------------------------------------------------------------
Ran 2 tests in 0.014s
FAILED (errors=1)
```



FAILED

Add the sort by string option

File sorter.py:

```
def sort_numbers(sort_by_string=False):
    sorted_list = []
    for line in file('to_sort.txt'):
        if sort_by_string:
            sorted_list.append(line.strip())
        else:
            sorted_list.append(int(line))
    sorted_list.sort()
    sorted_file = file('sorted.txt','w')
    for num in sorted_list:
        sorted_file.write('%s\n'%num)
```

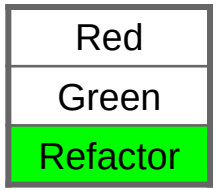
Red
Green
Refactor

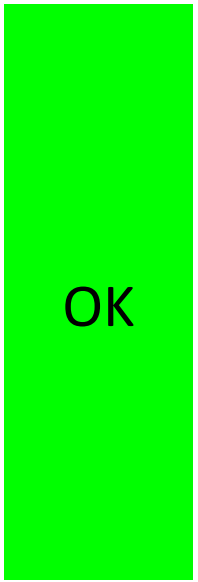
OK

Add the sort by string option

File sorter.py:

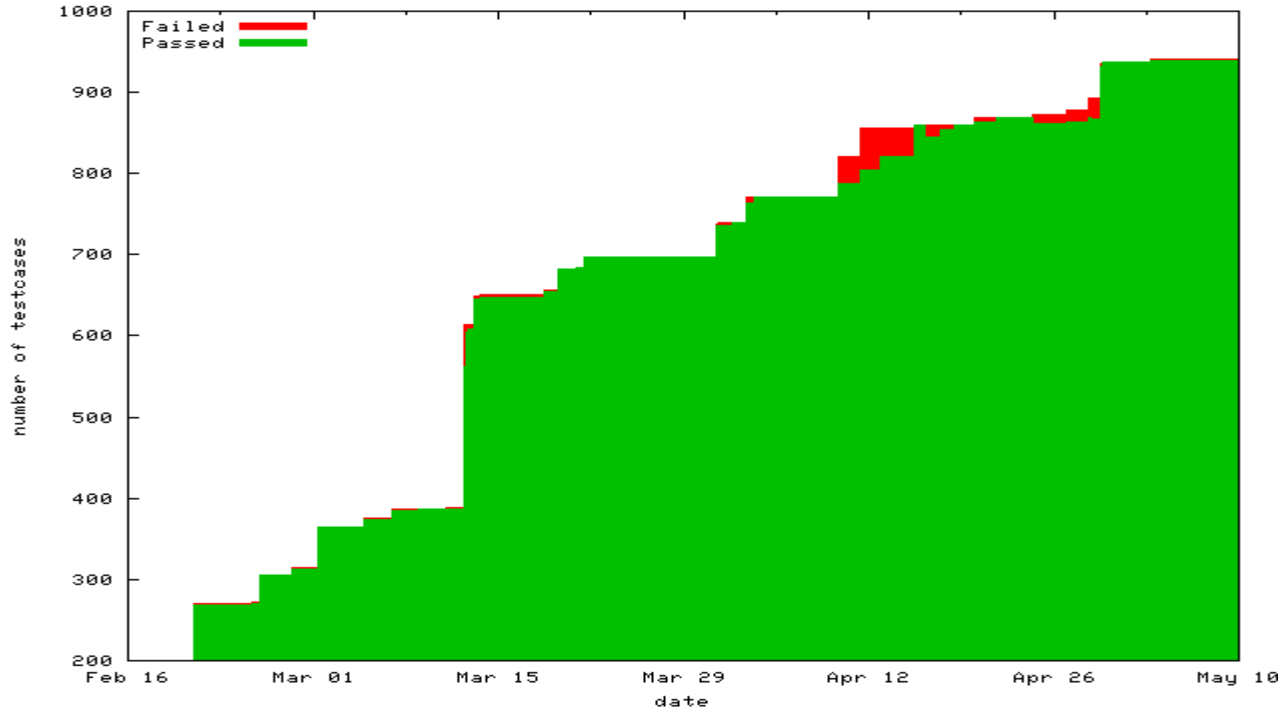
```
def sort_numbers(sort_by_string=False):
    sorted_list = []
    for line in file('to_sort.txt'):
        if sort_by_string:
            sorted_list.append(line.strip())
        else:
            sorted_list.append(int(line))
    sorted_list.sort()
    sorted_file = file('sorted.txt','w')
    for element in sorted_list:
        sorted_file.write('%s\n'%element)
```



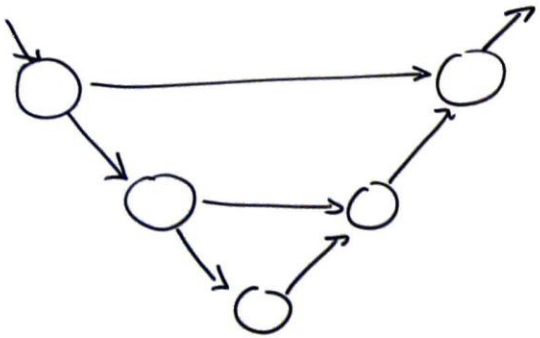


TDD was used, framework and facilities grew

Progress and Quality is measurable



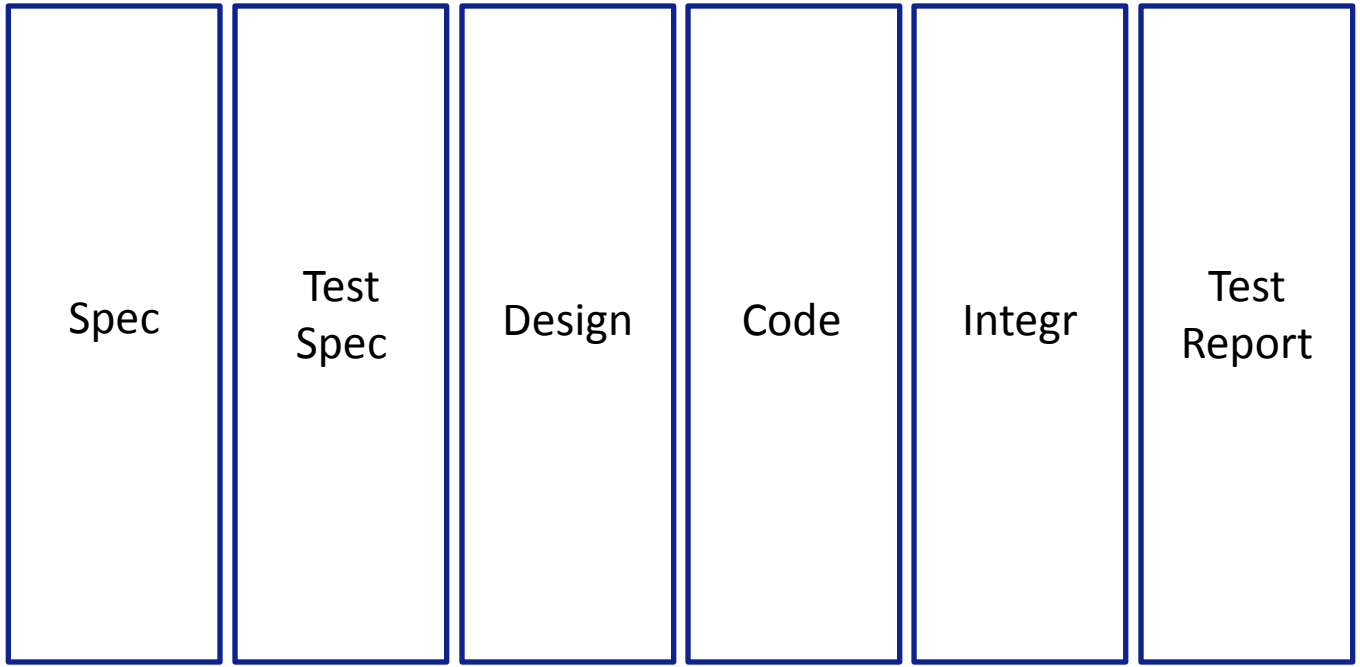
Taking Agile/TDD one step further: Traditional, Cowboy coding and Agile



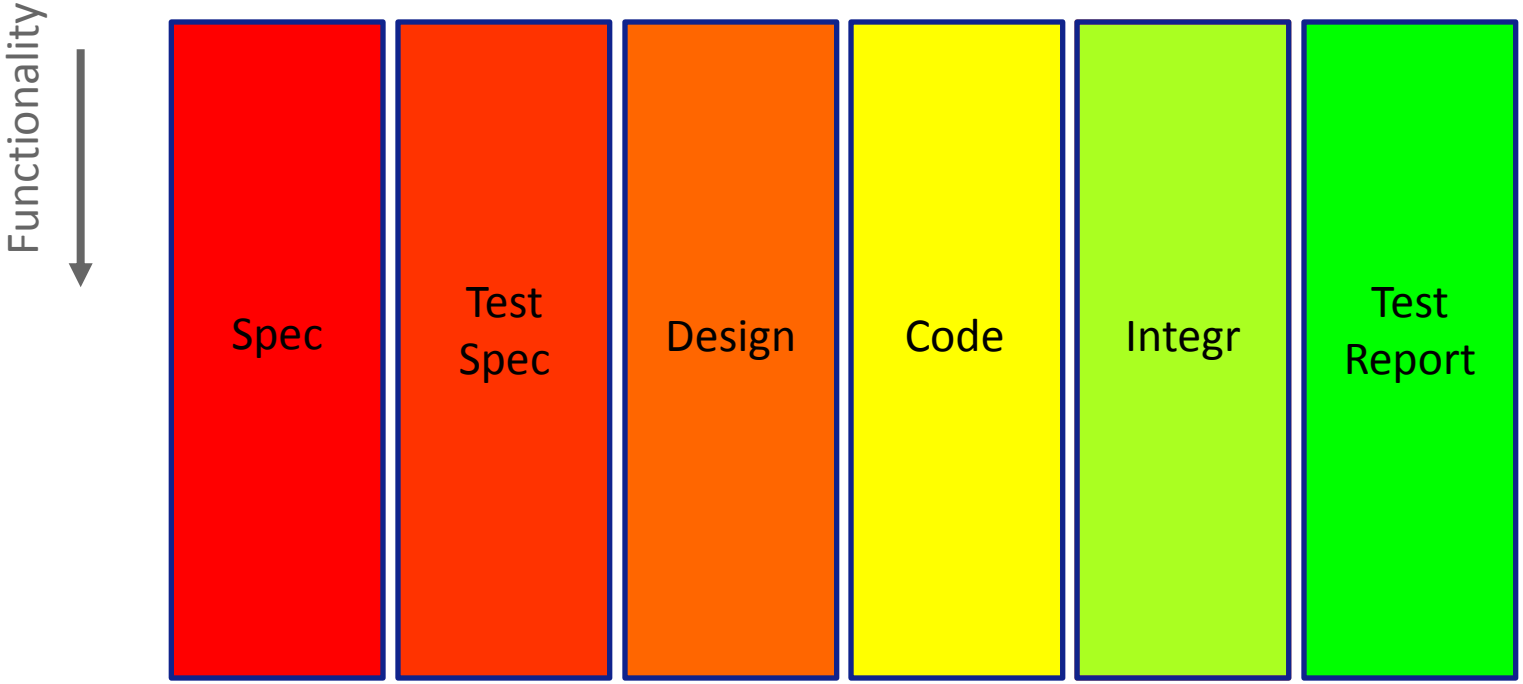
Stability

Flexibility

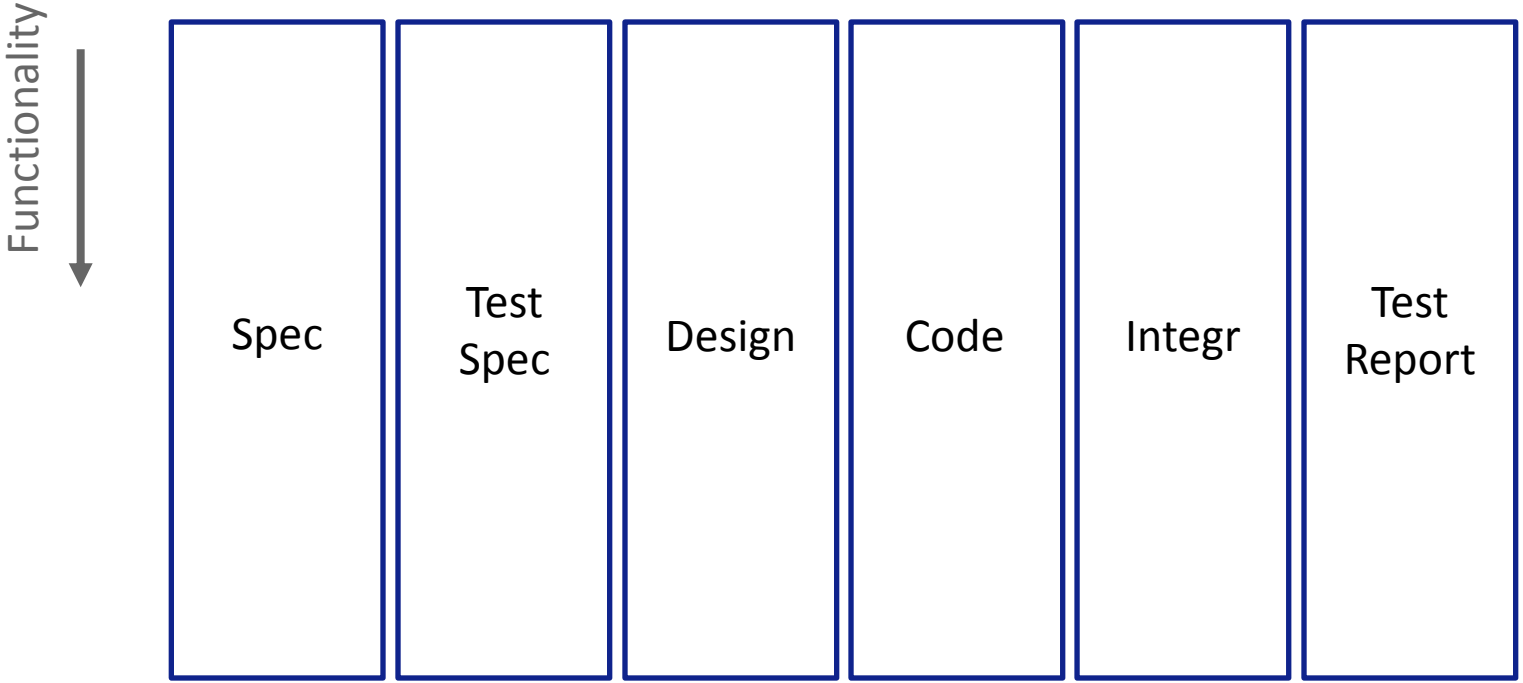
Traditional Software Component Development Process



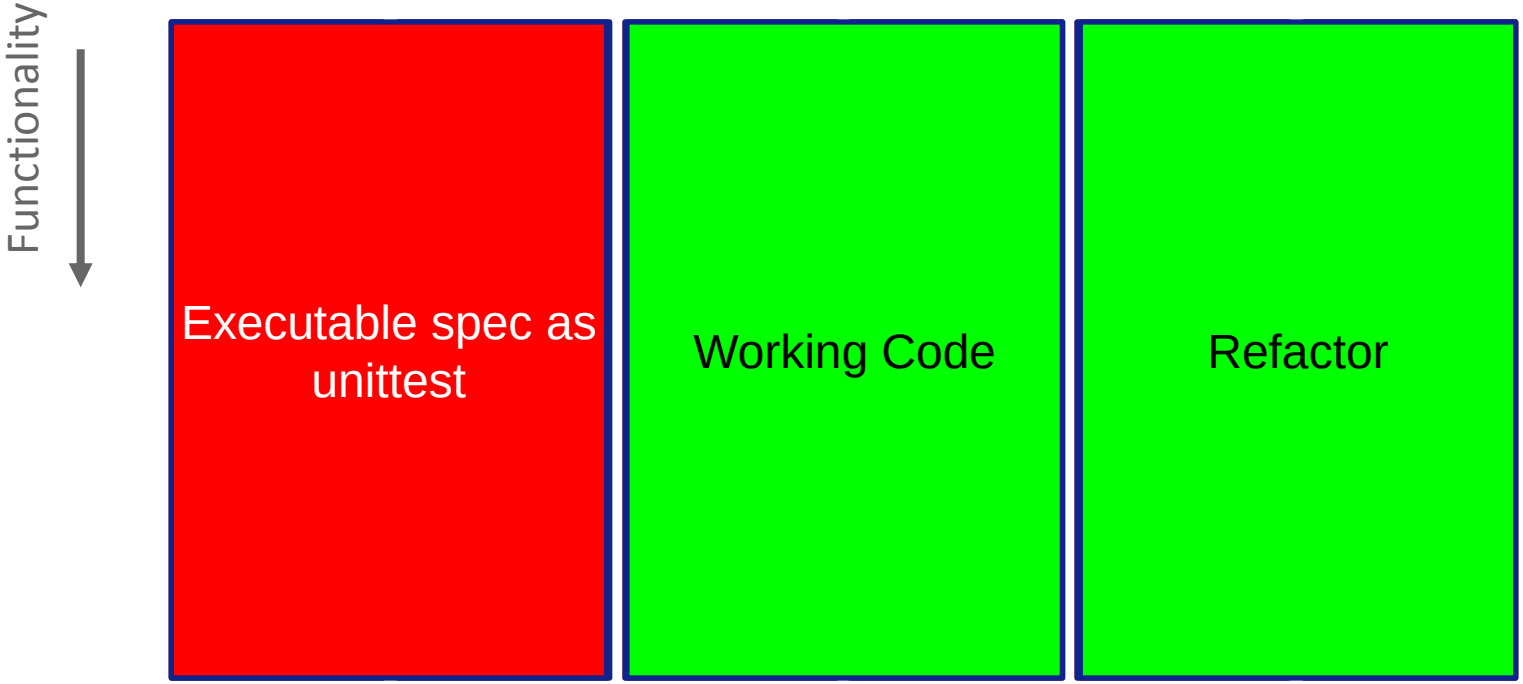
Traditional Software Component Development Process



Towards Agile Process: short iterations



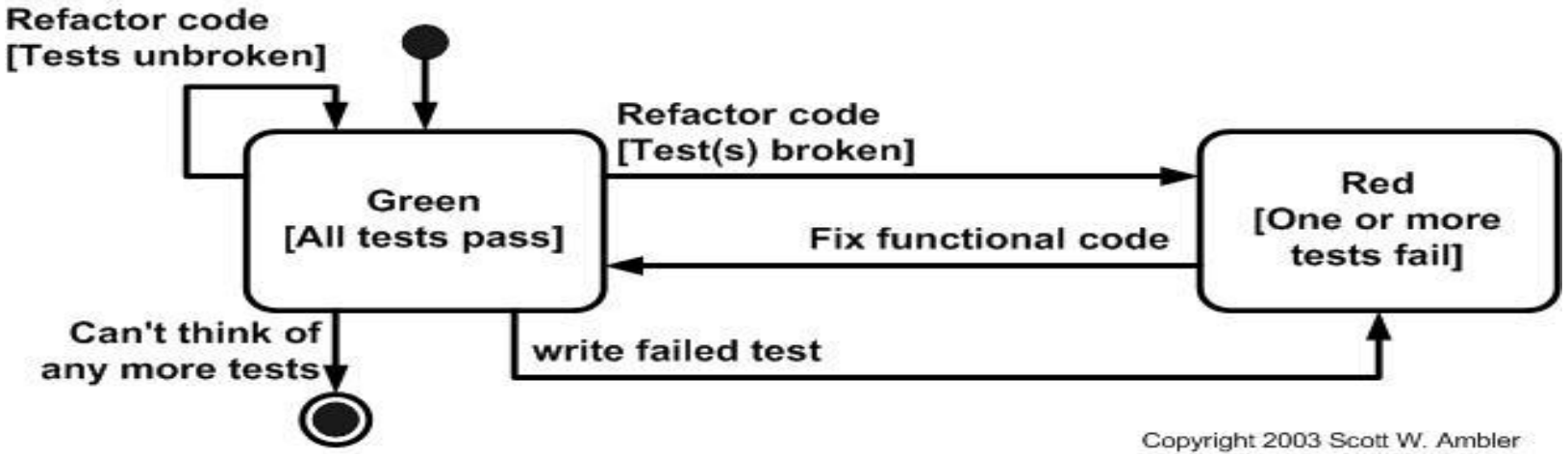
One step further towards TDD



Test Driven Development

Write tests first, then code

Three phases: Red, Green, Refactor



Management objections/fear

Objection

Skipping design

No documention

Change

Management objections/fear

Objection	Agile remedy
Skipping design	Continuous Design
No documention	Less documentation
Change	Don’t be afraid

Management objections/fear

Objection	Agile remedy	Pragmatic applied remedy
Skipping design	Continuous Design	Vision, High-level design doc
No documention	Less documentation	Doc generation+Lean docs
Change	Don’t be afraid	Just do it + Don’t emphasize

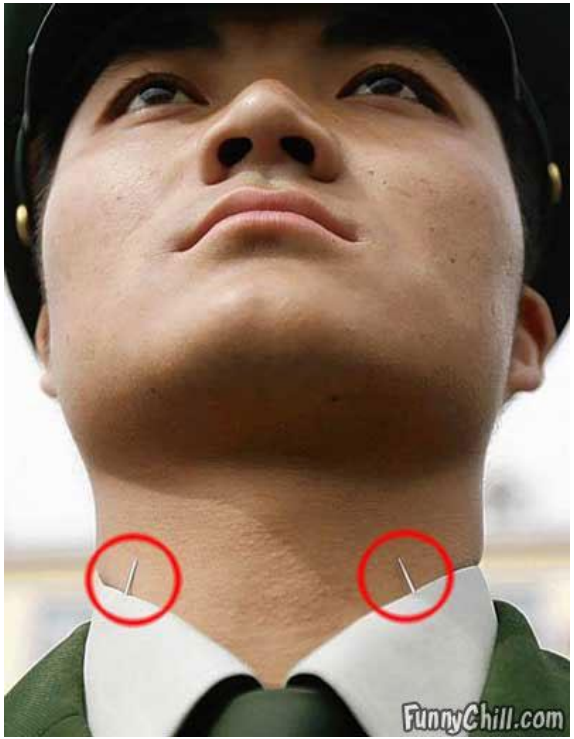
Internal obstacles: Discipline

Remedy:

- Pair programming
- Peer pressure
- Institutionalizing change

Pays off in the end,

but feels awkward at start



(Un)Satisfying feeling after using TDD

Traditional: feeling of not being fully complete

TDD at first: feeling stays, but there are no things to do

TDD eventually: trust, satisfying

Hurdle at first, blessing eventually

New functionality vs bugfixing

- Requirements are relatively easy for new functionality
- Difficult bugs, the investigation takes up almost all of the time
  TDD doesn’t help here

Test Driven Development Lessons learned

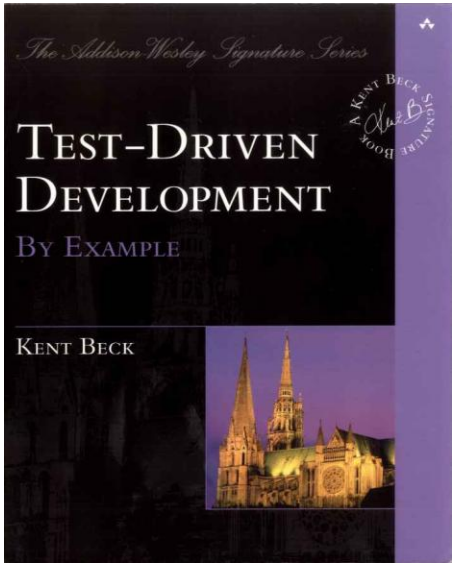
- TDD ≠ Test, but Development
- SMARTer requirements
- Flexible to changes
- Measurable progress
- Quality thru test focus

- Discipline
- TDD for functionality, difficult for bugfixing
- Document rationale requirement
- Document generation
- Test requirements, not design
- Design still needed
- Don’t change the whole world (at once)

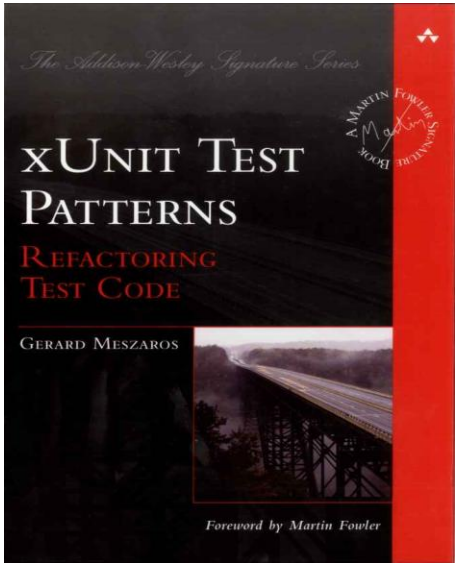
Just do it

Material available on TDD/unit tests

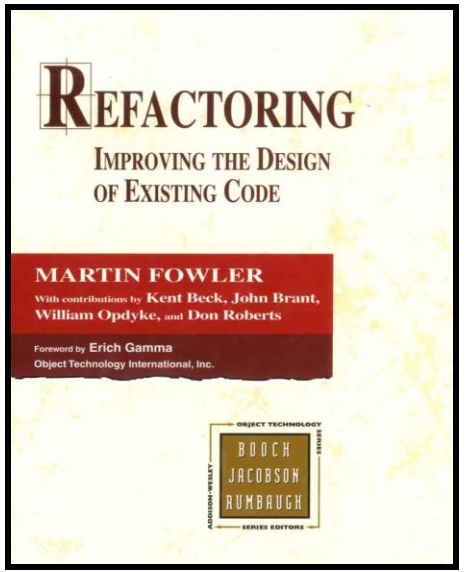
www.testdriven.com



www.xunitpatterns.com



www.refactoring.com



ASML